BAB II
LANDASAN TEORI

2.1 Korea

Korea merupakan sebuah semenanjung yang terletak diantara Cina dan Jepang. Awalnya semenanjung Korea merupakan satu negara utuh. Namun, setelah Perang Dunia ke II pada tahun 1945, semenanjung Korea terpisah menjadi dua negara, yaitu Republik Korea (Korea Selatan) dan Republik Demokratik Korea (Korea Utara).

Korea Selatan memiliki luas 99.274 km^2, lebih kecil dari luas Korea Utara Keadaan topografinya sebagian besar berbukit dan tidak rata. Pegunungan di wilayah timur umumnya menjadi hulu sungai-sungai besar, seperti sungai Han dan sungai Naktong. Sementara wilayah barat merupakan bagian rendah yang terdiri dari daratan pantai yang berlumpur.

2.1.1 Sejarah Korea

Pada 1 Masehi, terdapat tiga kerajaan yang mendominasi di semenanjung Korea. Ketiga kerajaan tersebut adalah Goguryo, Silla, dan Baekje. Ketiga kerajaan tersebut saling bersaing secara ekonomi dan militer. Goguryo merupakan kerjaan terkuat karena selalu bisa menangkis serangan-serangan dari sinasti-dinasti Tiongkok.

Silla merupakan kerajaan yang paling lemah dan paling terbelakang dari ketiga kerjaan tersebut. Namu, kerajaan yang terletak di bagian paling ujung tenggara semenanjung Korea ini secara geografis terlepas dari pengaruh Cina., sehingga Silla menjadi kerajaan yang lebih terbuka terhadap kebiasaan-kebiasaan dan serta ide-ide yang berasal dari Cina. Secara bertahap kerajaan tersebut menjadi kuat dan akhirnya bisa menundukkan Goguryo.

Silla runtuh pada abad kesembilan dan semenanjung Korea mulai didominasi oleh kerajaan baru, Dinasti Goryeo, yang merupakan cikal-bakal nama

Korea pada masa modern. Selama masa pemerintahan Dinasti Goryeo, hukum yang baru dibuat, pelayanan masyarakat dibentuk, serta penyebaran agama Buddha berkembang pesat.

Pada tahun 1392, Jenderal Yi Seong-gye mendirikan dinasti baru yang disebut Joseon. Para penguasa awal Dinasti Joseon mendukung ajaran Konfusianisme sebagai filsafat penuntun kerajaan, dengan tujuan melawan pengaruh Budha yang dominan selama masa pemerintahan Dinasti Goryeo.

Selama bertahtanya Raja Sejong yang Agung (1418-1450), yang merupakan raja keempat dari Dinasti Joseon, bangsa Korea menikmati masa berkembangnya kebudayaan dan kesenian yang belum pernah terjadi sebelumnya. Di bawah bimbingan Raja Sejong, kaum cendekia pada akademi kerajaan menciptakan alfabet Korea yang bernama Hangeul. Huruf ini kemudian dinamakan Hunminjeongeum, atau “sistem fonetik yang tepat untuk mendidik masyarakat.”

Pada abad ke-19, Korea merupakan ”Kerajaan Pertapa,”yang bersikeras untuk tidak menuruti permintaan-permintaan dunia barat untuk membangun hubungan diplomatik dan perdagangan.

Seiring berjalannya waktu, beberapa negara Asia dan Eropa yang memiliki ambisi-ambisi imperialistik bersaing satu dengan yang lain untuk meraih pengaruh atas Semenanjung Korea. Jepang, setelah menang perang melawan Cina dan Rusia, secara paksa menganeksasi Korea dan mendirikan pemerintahan kolonial pada tahun 1910.

Pemerintahan kolonial membangkitkan semangat patriotisme bangsa Korea. Kaum terdidik Korea dibuat marah oleh kebijakan resmi asimilasi yang diberlakukan oleh pemerintah Jepang, yang bahkan melarang pendidikan bahasa Korea di sekolah-sekolah Korea. Pada tanggal 1 Maret 1919, demonstrasi damai menuntut kemerdekaan menyebar ke seluruh wilayah Korea.

Aparat Jepang bertindak sangat keras terhadap para demonstran dan para pendukungnya serta membantai ribuan jiwa. Walaupun gagal, Gerakan

Kemerdekaan 1 Maret ini menciptakan ikatan yang kuat di antara rakyat Korea berkaitan dengan identitas nasional dan semangat patriotisme mereka.

Gerakan ini bermuara pada didirikannya Pemerintah Sementara di Shanghai, Cina, serta perjuangan bersenjata yang terorganisir melawan kaum kolonial Jepang di Manchuria. Gerakan Kemerdekaan ini masih diperingati oleh masyarakat Korea tiap tanggal 1 Maret, yang akhirnya diresmikan menjadi hari libur nasional.

Selama masa penjajahan, ekploitasi ekonomi Jepang terhadap Korea terus berlanjut. Kehidupan bangsa Korea memburuk selama masa penjajahan sampai akhir Perang Dunia II tahun 1945.

Dengan menyerahnya Jepang di tahun 1945, PBB membuat rencana administrasi bersama Uni Soviet dan Amerika Serikat, namun rencana tersebut tidak terlaksana. Pada tahun 1948, pemerintahan baru terbentuk, yang demokratik (Korea Selatan) dan komunis (Korea Utara) yang dibagi oleh garis lintang 38 derajat.

2.1.2 Alfabet Korea (*Han-Gul)*

Seluruh rakyat korea berbicara dan menulis dengan bahasa yang sama, yang menjadi factor penentu pembentukan identitas nasional. Bahasa Korea memiliki berbagai macam dialek selain dialek umum yang sering digunakan di Seoul dan daerah sentral.

Berdasarkan studi linguistik dan etnologi, bahasa Korea diklasifikasikan ke dalam keluarga bahasa Altaik, yang mencakup bahasa Turki, Mongol, Tibet, dan Jepang. Bahasa Korea cukup mirip dengan bahasa Jepang dari segi struktur tatabahasa, keduanya sama-sama memiliki kata serapan dari bahasa China.

Alfabet Korea dikenal dengan sebutan *Han-gul*, ditemukan pada abad ke 15 oleh Raja Agung Sejong. Sebelum alfabet ini terbentuk, masyarakat Korea menulis dengan karakter China, yang memiliki sistem bahasa yang amat sangat berbeda. Mempelajari tulisan China membutuhklan waktu yang sangat lama dan sulit. Hanya keturunan bangsawan yang bisa memahami sistem penulisan

tersebut. Hal ini menyebabkan angka buta huruf di Korea sangatlah besar pada masa itu.

Alfabet Korea terdiri dari 40 huruf, yaitu 10 vokal tunggal, 11 vokal rangkap, 14 konsonan tunggal, dan 5 konsonan ganda, yang dapat digunakan untuk berbagai macam kelomok suku kata. Alfabet ini sangat sederhana namun sistematis dan bersifat menyeluruh. *Han-gul* mudah untuk dipelajari dan dituliskan sehingga dengan sistem alphabet ini menumbuhkan angka melek huruf dan majunya industri penerbitan Korea.

Pembagian huruf Han-gul ini akan digambarkan dengan tabel seperti berikut:

Tabel 2.1 Vokal Tunggal

Vokal Tunggal		
Han-gul	Latin	Contoh Pelafalan
ㅏ	A	**a**yam
ㅓ	Ŏ	**o**tak
ㅗ	O	**o**rang
ㅜ	U	**u**dang
ㅡ	Ŭ	k**e**lakar
ㅑ	Ya	ba**ya**m
ㅕ	Yŏ	ko**yo**
ㅛ	Yo	**yo**yo
ㅠ	Yu	ba**yu**
ㅣ	I	**i**kan

Tabel 2.2 Vokal Gabungan

Vokal Gabungan		
Han-gul	Latin	Contoh Pelafalan
ㅐ	Ae	**e**lang
ㅔ	E	**e**nak
ㅚ	Oe	k**ue**
ㅒ	Yae	**y**en
ㅖ	ye	**y**eni
ㅟ	wi	**wi**sata
ㅢ	ŭi	**eu**leuh
ㅘ	wa	**wa**nita
ㅙ	wae	**we**sternisasi
ㅝ	wŏ	**wo**l
ㅞ	we	**we**sel

Tabel 2.3 Konsonan Tunggal

Konsonan Tunggal		
Han-gul	Latin	Contoh Pelafalan
ㅂ	b/p	**b**apak / **p**apaya
ㅈ	j/ch	**j**arum / **c**ela
ㄷ	d/t	**d**amar / **t**ulang
ㄱ	k/g	**k**amar / **g**igi
ㅅ	s	**s**uper

ㅁ	m	**m**anis
ㄴ	n	**n**ama
ㅇ	-/ng	tidak dibaca / aba**ng**
ㄹ	r/l	**r**usa / **l**ama
ㅎ	h	**h**ujan
ㅋ	k’	**k**ucing
ㅌ	t’	**t**ali
ㅊ	ch’	**c**inta
ㅍ	p'	**p**asang

Tabel 2.4 Konsonan Gabungan

Konsonan Gabungan		
Han-gul	Latin	Contoh Pelafalan
ㄲ	kk	**k**ambing
ㄸ	tt	**t**ali
ㅉ	tch	**c**acing
ㅆ	ss	**s**alam
ㅃ	pp	**p**anik

2.1.2.1 Penggabungan Huruf *Han-gul*

Huruf-huruf Han-gul pada dasarnya dibentuk dari penggabungan unsur vokal dan konsonan. Adapun contoh pembentukan dan posisi perletakkan masing-masing unsur dapat dilihat pada contoh di bawah ini:

Konsonan-Vokal

ㅂ	ㅏ	→	바
p	a		pa
ㅇ	ㅏ	→	아
-	a		a

Konsonan-Vokal-Konsonan (Akhir)

ㅂ	ㅏ	→	반
p	a		
ㄴ			Pan
n			

Berikut ini tabel kumpulan penggabungan huruf vokal dan konsonan:

Tabel 2.5 Penggabungan Huruf Vokal dan Konsonan

Konsonan dan Vokal	ㅏ a	ㅣ i	ㅜ u	ㅔ e	ㅓ ŏ	ㅗ o
ㅇ (tak berbunyi)	아 a	이 i	우 u	애 e	어 ŏ	오 o
ㅂ b	바 ba	비 bi	부 bu	배 be	버 bŏ	보 bo
ㅈ j / ch	자 ja	지 ji	주 chu	재 che	저 chŏ	조 cho

ㄷ d / t	다 da	디 di	두 du	대 de	더 dŏ	도 do
ㄱ k / g	가 ka	기 ki	구 gu	개 ke	거 kŏ	고 go
ㅅ s	사 sa	시 si	수 su	새 se	서 sŏ	소 so
ㅁ m	마 ma	미 mi	무 mu	매 me	머 mŏ	모 mo
ㄴ n	나 na	니 ni	우 nu	내 ne	너 nŏ	누 no
ㅊ ch’	차 ch’a	치 ch’i	추 ch’u	채 ch’e	처 ch’ŏ	추 cho

1. Partikel 은 / 는 (*ŭn / nŭn*)

Berfungsi sebagai penunjuk subjek dalam konteks untuk membandingkan atau menunjukkan suatu kekhasan dari subjek yang membedakannya dengan kata lain.

- Bila kata di depannya berakhiran konsonan, partikel yang digunakan adalah 은 (*ŭn).*
- Bila kata di depannya berakhiran vokal, partikel yang digunakan adalah 는 (*nŭn*)

2. Akhiran 이다 (*ida*)

Diletakkan di akhir kalimat dan akan mengalami perubahan sebagai berikut:

- Bila kata di depannya berakhiran konsonan, 이다 (*ida*) menjadi이에요 *(iyeyo)*
- Bila kata di depannya berakhiran vokal, 이다 (*ida*) menjadi 애요 *(yeyo)*

Pola kalimat berita sederhana (positif):

Subjek + 은 / 는 + Kata Benda + 가 / 이 + 이애요 / 애요
(*ŭn / nŭn*) *(ga /i)* *(iyeyo / yeyo)*

Contoh:

Saya adalah seorang murid.
나**는** 대학생이**에요**
*na**nŭn** daehaksaeng**iyeyo***

2.2 *Mobile Learning*

Mobile Learning merupakan sebuah istilah yang mengacu kepada suatu kegiatan, biasanya pendidikan atau pelatihan yang menggunakan beberapa jenis perangkat *mobile* (David Parson, 2007). Hal ini merupakan upaya dari pemanfaatan teknologi dari perangkat bergerak yang telah melebihi fungsi dasarnya yaitu untuk berkomunikasi. Selain itu perangkat bergerak ini juga telah mendapat dukungan dari segi konektifitasnya.

Mobile Learning sendiri merupakan bagian dari *e-Learning* (*Electronic Learning*), secara konsep *e-Learning* dan *mobile Learning* merupakan bagian dari d-learning (*Distance Learning*) (Georgiev, Georgieva, & Smrikarov, 2004).

Mobile learning didefinisikan oleh Clark Quinn (Quinn, 2000) sebagai : *The intersection of mobile computing and e-learning: accessible resources wherever you are, strong search capabilities, rich interaction, powerful support for effective learning, and performance-based assessment. E-learning independent of location in time or space*. Berdasarkan definisi tersebut, *mobile learning* merupakan model pembelajaran yang memanfaatkan teknologi informasi dan komunikasi. Pada konsep pembelajaran tersebut *mobile learning* membawa manfaat ketersediaan materi ajar yang dapat di akses setiap saat dan visualisasi materi yang menarik.

Untuk lebih memaksimalkan *mobile learning* ada beberapa kemampuan yang harus disediakan oleh perangkat pembelajaran *mobile learning* tersebut seperti kemampuan untuk terkoneksi dengan perangkat lain terutama komputer, kemampuan menyajikan informasi pembelajaran dan kemampuan untuk merealisasikan komunikasi bilateral antara pengajar dan pembelajar. *Mobile learning* adalah pembelajaran yang unik karena pembelajar dapat mengakses materi pembelajaran, arahan dan aplikasi yang berkaitan dengan pembelajaran, kapanpun dan dimanapun. Hal ini akan meningkatkan perhatian pada materi pembelajaran, membuat pembelajaran menjadi pervasif, dan dapat mendorong motivasi pembelajar kepada pembelajaran sepanjang hayat. Selain itu, dibandingkan pembelajaran konvensional, *mobile learning* memungkinkan adanya lebih banyak kesempatan untuk berinteraksi secara informal diantara pembelajar.

Karena *mobile learning* merupakan bagian dari *e-learning*, maka ada 2 tipe pembagian dari metode pembelajaran ini yaitu, (Efendi dan Zhuang, 2005):

1. *Synchronous training*

   Synchronous berarti “pada waktu yang sama”. Jadi *Synchronous training* adalah tipe pelatihan yang terjadi ketika pengajar sedang mengajar dan murid sedang belajar. Tipe ini lebih sering digunakan saat seminar atau konferensi atau yang lebih dikenal dengan kuliah *online*.

2. *Asynchronous training*

   Asynchronous berarti “tidak pada waktu yang bersamaan”. Jadi seseorang dapat mengambil pelatihan pada waktu yang berbeda dengan pengajar memberikan pelatihan. Pelatihan ini lebih populer di dunia *mobile learning* karena memberikan keuntunga lebih bagi peserta, karena dapat mengakses pelatihan kapanpun dan dimanapun.

2.2.1 Keuntungan *Mobile Learning*

Menurut Efendi dan Zhuang (2005), ada beberapa keuntungan dari penggunaan mobile learning, diantaranya sebagai berikut:

1. Biaya

   Penghematan biaya bisa dilakukan karena dapat menekan biaya untuk urusan teknis yang biasa digunakan seperti pembelajaran konvensional, seperti penyediaan peratalan tuli, papan, konsumsi untuk pengajar, proyektor, dan lainnya.

2. Fleksibilitas waktu dan tempat

   Mobile learning dapat membuat penggunanya menyesuaikan waktu dan tempat belajar. Mereka bisa menyisipkan pembelajaran disaat waktu luang dan tempat yang berbeda.

3. Standarisasi pengajaran

   Adanya perbedaan kemampuan dalam memberikan pengajaran oleh guru atau pengajar menyebabkan peserta memiliki perbedaan dalam menyerap pembelajaran, terkadang standar pengajaran juga tergantung suasana hati pengajar. *Mobile learning* dapat menghapus perbedaan tersebut, pelajaran di mobile learning memiliki kualitas yang sama setiap kali diakses dan tidak tergantung suasana hati pengajar.

4. Fleksibilitas kecepatan pembelajaran

   Setiap pelajar memiliki kemampuan yang berbeda dalam menyerap pelajaran, ada yang cepat dan ada yang lambat. Hal ini bisa diatasi oleh mobile learning, karena kecepatan belajar tergantung dari masing-masing pelajar.

2.2.2 Keterbatasan Mobile Learning

Walaupun *mobile learning* menawarkan beberapa keuntungan bagi penggunanya. Praktik ini juga memeliki beberapa keterbatasan yang harus diwaspadai oleh pengelola *mobile learning*, berikut beberapa keterbatasan dalam mobile learning (Efendi dan Zhuang, 2005):

1. Budaya

   Sebagian orang merasa tidak nyaman mengikuti sistem *mobile learning*. Penggunaan *mobile learning* menuntut budaya *self-learning,* dimana seeorang memotivasi diri sendiri agar mau belajar. Sebaliknya, pada sebagian besar budaya pendidikan Indonesia, motivasi belajar banyak tergantung dari pengajar. Disini kita juga harus melihat kebiasaan penggunaan teknologi dari pelajar. Apabila mereka tidak terbiasa menggunaka perangkat mobile, maka penerapan mobile learning akan memakan waktu yang lama.

2. Investasi

   Meskipun *mobile learning* menghemat banyak biaya, tapi suatu institusi harus mengeluarkan investasi awal yang cukup besar untuk memulai penerapan mobile learning. Investasi dapat berupa biaya perancangan dan pembuatan program *learnig management sistem,* paket pembelajaran, promosi dan lainnya.

3. Teknologi

   Karena teknologi yang digunakan beragam ada kemungkinan teknologi tersebut tidak sejalan dengan yang sudah ada dan terjadi konfilk teknologi sehingga mobile learning tidak berjalan dengan baik.

4. Infrastruktur

   Mobile learning tidak lepas dari segi konektifitas, salah satunya akses internet yang berguna untuk mengakses konten yang ada di*server*. Belum meratanya layanan koneksi internet menjadi sebuah keterbatasan dalam penerapan *mobile learning*.

5. Materi

   Tidak semua materi bisa dimasukan dalam *mobile learning*, beberapa materi memerlukan adanya kegiatan fisik, seperti olahraga, instrumen musik, seni rupa, tari dan lainnya

2.3 Android

Android merupakan sebuah sistem sistem operasi untuk perangkat *mobile* berbasis *linux* yang meliputi sistem operasi, *middleware*, dan aplikasi yang dirilis oleh Google. Sedangkan Android SDK (*Software Development kit*) menyediakan Tools dan API (*Application Programming Interface*) yang diperlukan untuk mengembangkan aplikasi pada *platform* Android dengan menggunakan bahasa pemrograman Java (Mulyadi, 2010). Android dikembangkan oleh Google bersama OHA (*Open Handset Alliance*) yaitu aliansi perangkat selular terbuka yang terdiri dari 47 perusahaan *Hardware, Software* dan perusahaan telekomunikasi ditujukan untuk mengembangkan standar terbuka bagi perangkat selular.

2.3.1 Sejarah Android

Pada tahun 2005 Google mengakusisi Android Inc yang pada saat itu dimotori oleh Andy Rubin, Rich Miner, Nick Sears dan Chris White. Kemudian pada tahun itu juga memulai membangun platform Android secara intensif.

Pada saat perilisan perdana *Android*, 5 November 2007, *Android* bersama OHA menyatakan mendukung pengembangan *open source* pada perangkat *mobile*. Di lain pihak, *Google* merilis kode-kode *Android* dibawah lisensi

Apache, sebuah lisensi perangkat lunak dan *open source platform* perangkat seluler.

Di dunia ini terdapat dua jenis distributor *Android*, yang pertama yang mendapat dukungan penuh dari *Android* atau *Android* Mail *Service*s (GMS) dan kedua adalah yang benar-benar bebas ditribusinya tanpa dukungan langsung *Google* atau dikenal dengan Open Handset Distribution (OHD).

Sekitar september 2007 *Google* mengenalkan Nexus One, Salah satu jenis *Smartphone* yang menggunakan *Android* sebagai sistem operasinya. Telepon seluler yang di produksi oleh HTC Corporation dan tersedia di pasaran pada 5 januari 2010. Pada 9 Desember 2008, diumumkan anggota baru yang bergabung dalam program kerja *Android* ARM Holdings, Atheros, Softbank, Sony Ericson, Toshiba Corp, dan Vodafone Gropu Plc. Seiring pembentukan Open Handset Aliance, OHA mengumumkan produk perdana mereka, *Android*, perangkat *mobile* yang merupakan modifikasi kernel linux 2.6. Sejak *Android* di rilis telah dilakukan berbagai pembaruan berupa perbaikan bug serta fitur baru.

Pada saat ini, sebagian besar Vendor-Vendor *Smartphone* sudah memproduksi *Smartphone* berbasis *Android*, Vendor-vendor itu antara lain HTC, Motorola, Samsung, LG, HKC, Huawei, Archos,*Web*station Chamangi, Dell, Nexus, SciPhone, WayteQ, Sony Ericson, LG, Acer, Philips, T-*Mobile*, Nexian, IMO, Asus dan masih banyak lagi vendor *smartphone* di dunia yang memproduksi *Android*. Hal ini, karena *Android* itu adalah sistem operasi yang *open source* sehingga bebas didistribusikan dan dipakai oleh vendor manapun. Bukan hanya menjadi sistem operasi pada *smartphone*, saat ini *Android* menjadi pesaing utama Apple pada sistem operasi Tablet PC. Pesatnya pertumbuhan *Android* selain faktor yang disebutkan diatas adalah karena *Android* itu sendiri merupakan *platform* yang sangat lengkap baik itu sistem operasinya, Aplikasi dan Tool pengembangan, Market aplikasi *Android* serta dukungan yang sangat tinggi dari komunitas *open source* di dunia, sehingga *Android* terus berkembang baik dari segi teknologi maupun dari segi jumlah *device* di seluruh dunia.

Berikut adalah beberapa sistem operasi yang telah dirilis oleh *Android* hingga saat ini:

2.3.1.1 *Android* Versi 1.1

Pada 9 Maret 2009, *Google* merilis *Android* Versi 1.1. *Android* versi ini dilengkapi dengan pembaruan pada aplikasi, jam alarm, *voice search* (pencarian suara), pengiriman pesan dengan *Gmail* dan pemberitahuan *email*.

2.3.1.2 *Android* Versi 1.5 (*Cupcake*)

Pada versi ini terdapat beberapa pembaruan termasuk juga penambahan beberapa fitur dalam versi ini, yakni kemampuan merekam dan menonton video dengan modus kamera, dukungan *bluetooth* A2DP, kemampuan terhubung secara otomatis ke *headset Bluetooth*, animasi layar, dan *keyboard* pada layar yang dapat disesuaikan dengan sistem. Dirilis pada pertengahan Mei 2009.

2.3.1.3 *Android* Versi 1.6 (*Donut*)

Donut dirilis pada september 2009 dengan menampilkan proses pencarian yang lebih baik dibanding sebelumnya, penggunaan baterai indikator dan kontrol applet VPN. Fitur lainnya adalah galeri yang memungkinkan pengguna untuk memilih foto yang akan dihapus; kamera, *camcorder*, dan galeri yang diintegrasikan; CDMA/EVDO, VPN, *Gestures* dan *Text-to-Speech engine*; kemampuan *dial contact;* pengadaan resolusi WVGA.

2.3.1.4 *Android* versi 2.0/2.1 (*Eclair*)

Android ini diluncurkan pada 3 Desember 2009. Dilakukan perubahan, yaitu pengoptimalan *hardware*, perubahan *user interface* (UI) dengan *browser* baru dan dukungan HTML 5, daftar kontak yang baru,peningkatan *Google* maps 3.1.2, dukungan flash untuk kamera 3,2MP, digital *zoom* dan *Bluetooth* 2.1.

2.3.1.5 *Android* versi 2.2 (*Froyo*)

Pada 20 Mei 2010 kembali diluncurkan ponsel *Android* dengan versi 2.2 (*Froyo*) perubahan yang dilakukan meliputi optimasi kecepatan, memori, dan kinerja sistem operasi secara keseluruhan, dukungan *Adobe flash* 10.1 serta fungsi USB tethering maupun *Wi-fi hotspot*.

2.3.1.6 *Android* versi 2.3 (*Gingerbread*)

1 Desember 2010 *Google* kembali meluncurkan versi terbaru yaitu *Android* versi 2.3. Pada versi ini terdapat peningkatan manajemen daya, kontrol melalui aplikasi, penggunaan *multiple* kamera, peningkatan performa serta penambahan sensor seperti *gyroscope*.

2.3.1.7 *Android* Versi 3.0/3.1 (*Honeycomb*)

Versi ini berbeda dengan versi-versi sebelumnya. Versi ini dirancang khusus untuk PC Tablet sehingga memiliki *user interface* yang berbeda dan mendukung ukuran layar yang lebih besar. Selain itu, pada versi ini memungkinkan penggunaan multiprosesor dan akselerasi perangkat keras untuk grafis. SDK versi pertama diluncurkan pada februari 2011

2.3.1.8 *Android* Versi 4.0 (*Ice cream sandwich*)

Android 4.0 memberikan UI yang halus terpadu untuk ponsel dan tablet dan memperkenalkan fitur inovatif bagi pengguna dan pengembang. Versi memberikan banyak fitur baru dan teknologi yang membuat *Android* 4.0 yang sederhana, indah, dan semakin cerdas.

Difokuskan pada perubahan tampilan layar sentuh, *Android* 4.0 membuat tindakan umum lebih terlihat dan memungkinkan pengguna melakukan navigasi sederhana dengan gerakan intuitif. Animasi halus dan respon cepat seluruh sistem membuat interaksi mengikat dan menarik. Layar sentuh yang baru dioptimalkan

untuk layar dengan revolusi tinggi, meningkatkan keterbacaan dan membuat kesan baru yang lebih moderen dan elegan.

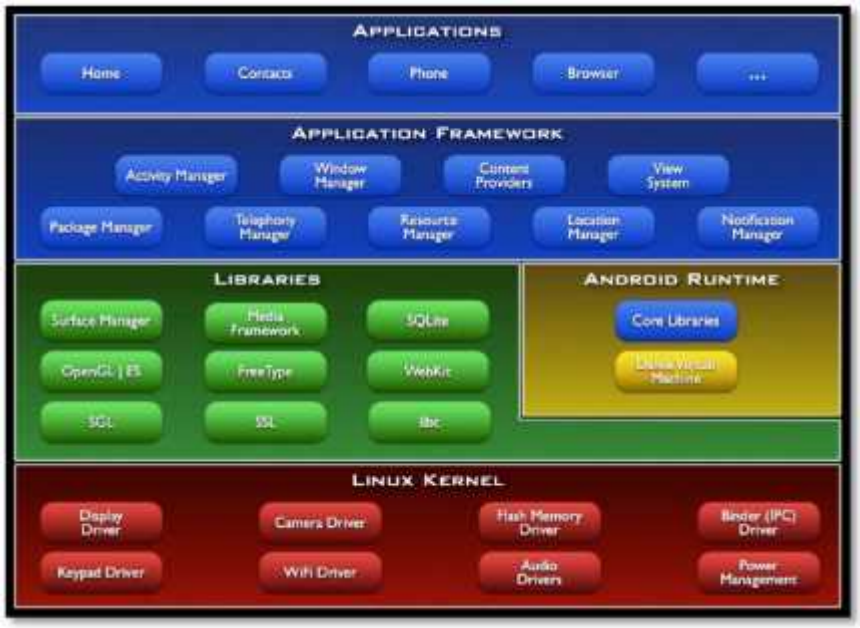
2.3.1.9 *Android* versi 4.1 (*Jellybean*)

Android 4.1 merupakan versi tercepat dan paling halus dari versi *android* yang lainnya. Terdapat banyak perbaikan pada seluruh *platform* dan menambahkan fitur-fitur baru yang besar bagi pengguna dan pengembang.

Android 4.1 dioptimalkan untuk memberikan kinerja *Android* terbaik dan *latency* sentuh terendah, tanpa menghabiskan banyak tenaga dan berintuisi dalam UI. Untuk memastikan *framerate* yang konsisten, *Android* 4.1 menambahkan waktu *vsync* di semua gambar dan animasi yang dilakukan oleh kerangka *Android*. Semuanya berjalan berbaris terhadap 16 *vsync* milidetik detak jantung. Aplikasi rendering, peristiwa sentuhan, komposisi layar, dan pembaharuan penampilan sehingga *frame* tidak akan berpindah kedepan atau kebelakang.

2.3.2 Arsitektur Android

Dalam paket sistem operasi Android terdiri dari beberapa unsur seperti tampak pada gambar 2.1. Secara sederhana arsitektur Android merupakan sebuah kernel Linux dan sekumpulan pustaka C / C++ dalam suatu *framework* yang menyediakan dan mengatur alur proses aplikasi.



Gambar 2.1 Arsitektur Android

2.3.2.1 *Linux Kernel*

Android bukan *linux*, akan tetapi android dibangun diatas *linux kernel* versi 2.6 yang kehandalannya sudah teruji. Untuk inti sistem servis *linux* yang digunakan seperti keamanan, manajemen momori, proses manajemen, *network* dan *driver* model. Seperti yang terlihat di gambar 2.6, linux kernel menyediakan *Driver* layar, *driver* kamera, flash memori, IPC (*Interprocces Communication*) untuk mengatur aplikasi dan keamanan, *driver* keypad, *driver* WiFi, *driver* audio, dan power management. *Kernel* juga bertindak sebagai lapisan abstrak antara *hardware* dan *software* stacknya (http://developer.android.com).

2.3.2.2 *Libraries*

Android menyertakan linbraries C / C++ yang digunakan oleh berbagai komponen dari sistem Android. Kemampuan ini disediakan kepada *developer* aplikasi melalui *framework* aplikasi Android. Beberapa inti libraries tercantun dibawah ini (Mulyadi, 2010):

1. *System C library*

   Variasi dari implementasi BSD (*Barkeley Software Distribution*) berasal dari sistem standar C *library* (*libc*), sesuai untuk perangkat *embedded* berbasis linux.
2. *Media libraries*

   Untuk merekam dan memutar berbagai format *Audio* dan *Video.*
3. *Surface Manager*

   Mengelola akses ke subsistem layar, lapisan komposit 2D dan grafis 3D dari beberapa aplikasi.
4. *LibWebCore*

   Librari untuk mesin web pada browser Android.
5. *SGL*

   Merupakan library untuk proses mesin grafis 2D.
6. *3D libraries*

   Digunakan untuk proses gambar yang membutuhkan daya 3D.
7. *FreeType*

   Merupakan library untuk bitmap dan vektor font rendering
8. *SQLite*

   Merupakan library mesin database yang kuat dan ringan, dan peghubung aplikasi yang tersedia.

2.3.2.3 *Android Runtime*

Pada Android tertanam paket pustaka inti yang menyediakan sebagian besar fungsi Android. Inilah yang membedakan Android dibandingkan dengan sistem operasi lain yang juga mengimplementasikan Linux. *Android Runtime* merupakan mesin virtual yang membuat aplikasi Android menjadi lebih tangguh dengan paket pustaka yang telah ada. Dalam *Android Runtime* terdapat 2 bagian utama,diantaranya (http://developer.android.com):

1. Pustaka Inti, Android dikembangkan melalui bahasa pemrograman Java, tapi *Android Runtime* bukanlah mesin *virtual* Java. Pustaka inti Android

menyediakan hampir semua fungsi yang terdapat pada pustaka Java serta beberapa pustaka khusus Android.

2. Mesin Virtual Dalvik, Dalvik merupakan sebuah mesin virtual yang dikembangkan oleh Dan Bornstein yang terinspirasi dari nama sebuah perkampungan yang berada di Iceland. Dalvik hanyalah interpreter mesin virtual yang mengeksekusi file dalam format *Dalvik Executable* (*.dex). Dengan format ini Dalvik akan mengoptimalkan efisiensi penyimpanan dan pengalamatan memori pada file yang dieksekusi. Dalvik berjalan diatas kernel Linux 2.6, dengan fungsi dasar seperti *threading* dan manajemen memori yang terbatas.

2.3.2.4 *Applications Frameworks*

Android merupakan "*Open Development Platform*" dimana Android menawarkan kepada pengembang untuk membangun aplikasi yang bagus dan inovatid. Pengembang bebas untuk mengakses perangakat keras, akses informasi resources, menjalankan service background, mengatur alarm, menambah status pemberitahuan dan lainnya. Pengembang memiliki akses penuh menuju API framework sperti yang dilakukan oleh aplikasi yang kategori inti.

Applications Frameworks merupakan layer dimana para pembuat aplikasi melakukan pengembangan aplikasi yang akan dijalankan di sistem operasi android. Komponen-komponen yang termasuk dalam *Applications Frameworks* adalah sebagai berikut (http://developer.android.com):

1. *Views*
2. *Content Provider*
3. *Resource Manager*
4. *Notification Manager*
5. *Activity Manager*

2.3.2.5 *Applications* dan *Widgets*

Applications dan *Widgets* ini adalah layer yang berhubungan dengan aplikasi saja. Biasanya aplikasi di download kemudian dilakukan instalasi dan jalankan aplikasi tersebut. Di layer ini terdapat aplikasi inti termasuk klien email, program SMS, kalender, peta, browser, kontak, dan lain-lain.

2.3.3 Android SDK (*Software Development Kit*)

Android SDK adalah tools API (*Application Programming Interface*) yang diperlukan untuk memulai pengembangan aplikasi pada *platform* Android menggunakan bahasa pemrograman Java. Android merupakan subset perangkat lunak untuk ponsel yang meliputi sistem operasi, *middleware* dan aplikasi kunci yang dirilis oleh Google. Untuk sumber SDK Android dapat dilihat dan diunduh langsung kesitus resmi Android di http://developer.android.com.

2.3.4 Komponen Aplikasi

Aplikasi Android ditulis dalam bahasa pemograman Java. Kode Java dikompilasi bersama dengan data file resource yang dibutuhkan oleh aplikasi, prosesnya di paket oleh *tools* yang dinamakan *atp tools* kedalam paket Android, sehingga menghasilkan file dengan ekstensi *apk.* Ada 4 jenis komponen pada aplikasi Android, yaitu (http://developer.android.com):

1. *Activities*

   Suatu *Activity* akan menyajikan *user interface* (UI) kepada pengguna, sehingga pengguna dapat melakukan interaksi. Pada umumnya sebuah aplikasi Android memiliki banyak *activity*, tergantung pada tujuan aplikasi dan desain dari aplikasi tersebut. Satu *activity* biasanya akan dipakai untuk menampilkan aplikasi atau yang bertindak sebagai *user interface* (UI) saat aplikasi diperlihatkan kepada *user*. Untuk pindah dari satu *activity* ke *activity* lain, kita dapat melakukan dengan satu even, misalnya klik tombol, memilih opsi atau menggunakan triggers tertentu. Secara hirarki sebuah windows

activity dinyatakan dengan *method* `Activity.setContentView()`. `ContentView` adalah objek yang berada pada *root hirarki.*

2. *Service*

   Service tidak memiliki *Graphic User Interface* (GUI), tetapi service berjalan secara background. Komponen *service* diproses tidak terlihat, memperbaharui seumber data dan menampilkan notifikasi. *Service* digunakan untuk melakukan pengolahan data yang terus diproses, bahkan ketika activiti tidak aktif

3. *Broadcast Receiver*

   Broadcasr receiver berfungsi menerima dan bereaksi untuk menyampaikan notifikasi. *Broadcast receiver* tidak memiliki user interface, tetapi memiliki sebuah activity untuk merespon informasi yang diterima atau menggunakan notification manager untuk memberi tahu pengguna, seperti lampu latar atau getaran.

4. *Content Provider*

   Content provider membuat kumpulan aplikasi data secara spesifik sehingga bisa digunakan oleh aplikasi lain. Data disimpan dalam file seperti database SQLite. *Content provider* menyediakan cara untuk mengakses data yang dibutuhkan oleh suatu activity, misalnya ketika kita menggunakan aplikasi yang membutuhkan peta atau aplikasi yang membutuhkan untuk mengakses data kontak dan navigasi, maka disinilah fungsi dari content provider.

2.3.5 Kelebihan *Platform* Android

Persaingan *platform* atau sistem operasi semankin ketat, ini dapat dilihat dari banyaknya sistem operasi yang ada seperti, Symbian, iPhone, Windows Mobile, BlackBerry, Java Mobile Edition, Linux Mobile (LiM0), dan banyak lagi. Namun ada beberapa hal yang menjadi kelebihan Android. Walaupun beberapa fitur-fitur yang ada telah muncul sebelumnya pada platform lain, Android adalah yang pertama menggabungkan hal seperti berikut (Amiral, 2010) :

1. Keterbukaan, Bebas pengembangan tanpa dikenakan biaya terhadap sistemkarena berbasiskan Linux dan *open source*. Pembuat perangkat

menyukai hal ini karena dapat membangun *platform* yang sesuai yang diinginkan tanpa harus membayar *royality*. Sementara pengembang *software* menyukai karena Android dapat digunakan diperangkat manapun dantanpa terikat oleh vendor manapun.

2. Arsitektur komponen dasar Android terinspirasi dari teknologi internet *Mashup*. Bagian dalam sebuah aplikasi dapat digunakan oleh aplikasi lainnya, bahkan dapat diganti dengan komponen lain yang sesuai dengan aplikasi yang dikembangkan.
3. Banyak dukungan *service*, kemudahan dalam menggunakan berbagai macam layanan pada aplikasi seperti penggunaan layanan pencarian lokasi, database SQL, browser dan penggunaan peta. Semua itu sudah tertanam pada Android sehingga memudahkan dalam pengembangan aplikasi.
4. Siklus hidup aplikasi diatur secara otomatis, setiap program terjaga antara satu sama lain oleh berbagai lapisan keamanan, sehingga kerja sistem menjadi lebih stabil. Pengguna tak perlu kawatir dalam menggunakan aplikasi pada perangkat yang memorinya terbatas.
5. Dukungan grafis dan suara terbaik, dengan adanya dukungan 2D grafis dan animasi yang diilhami oleh *Flash* menyatu dalam 3D menggunakan *OpenGL* memungkinkan membuat aplikasi maupun game yang berbeda.
6. Portabilitas aplikasi, aplikasi dapat digunakan pada perangkat yang ada saat ini maupun yang akan datang. Semua program ditulis dengan menggunakan bahas pemrograman Java dan dieksekusi oleh mesin virtual Dalvik, sehingga kode program portabel antara ARM, X86, dan arsitektur lainnya. Sama halnya dengan dukungan masukan seperti penggunaan *Keyboard*, layar sentuh, *trackball* dan resolusi layar semua dapat disesuaikan dengan program.

2.4 Analisa dan Perancangan Berorientasi Objek

Teknologi objek menganalogikan sistem aplikasi seperti kehidupan nyata yang didominasi oleh objek. Didalam membangun sistem berorientasi objek akan menjadi lebih baik apabila langkah awalnya didahului dengan proses analisis dan perancangan yang berorientasi objek. Tujuannya adalah untuk mempermudah

programmer didalam mendesain program dalam bentuk objek-objek dan hubungan antar objek tersebut untuk kemudian dimodelkan dalam sistem nyata (A.Suhendar, 2002).

Perusahaan *software*, Rational Software, telah membentuk konsorsium dengan berbagai organisasi untuk meresmikan pemakaian *Unified Modelling Language* (UML) sebagai bahasa standar dalam *Object Oriented Analysis Design* (O*OAD).*

2.4.1 *Unified Modelling Language* (UML)

Unified Modelling Language (UML) adalah sebuah bahasa yang telah menjadi standar dalam industri untuk visualisasi, merancang dan mendokumentasikan sistem piranti lunak. UML menawarkan sebuah standar untuk merancang model sebuah sistem (Dharwiyanti dan Wahono, 2006).

Untuk merancang sebuah model, UML memiliki beberapa diagram antara lain : *use case diagram, class diagram, statechart diagram, activity diagram, sequence diagram, collaboration diagram, component diagram, deployment diagram.*

2.4.1.1 *Use Case Diagram*

Use case diagram merupakan sebuah gambaran fungsionalitas sebuah sistem. Sebuah use case merepresentasikan interaksi antara aktor dengan sistem. *Use case* merupakan sebuah pekerjaan tertentu, misalnya login ke sistem, *create* sebuah daftar belanja, dan sebagainya. Seorang/sebuah aktor adalah sebuah entitas manusia atau mesin yang berinteraksi dengan sistem untuk melakukan pekerjaan-pekerjaan tertentu Dharwiyanti (2006).

Dalam sebuah sistem *use case diagram* akan sangat membantu dalam hal menyusun *requirment*, mengkomunikasikan rancangan dengan klien dan merancang *test case* untuk semua fitur yang ada pada sistem.

2.4.1.2 *Class Diagram*

Class adalah sebuah spesifikasi yang jika diinstansiasi akan menghasilkan sebuah objek danmerupakan inti dari pengembangan dan desain berorientasi objek. *Class* menggambarkan keadaan(atribut/properti) suatu sistem, sekaligus menawarkan layanan untuk memanipulasi keadaan tersebut (metoda/fungsi) Dharwiyanti (2006).

Class diagram menggambarkan struktur dan deskripsi *class, package* dan objek beserta hubungan satu sama lain seperti *containment*, pewarisan, asosiasi, dan lain-lain. *Class* memiliki tiga area pokok yaitu, nama, *stereotype* ,atribut dan metoda

2.4.1.3 *Sequence Diagram*

Sequence diagram menggambarkan interaksi antar objek di dalam dan di sekitar sistem (termasuk pengguna, *display*, dan sebagainya) berupa *message* yang digambarkan terhadap waktu. *Sequence diagram* terdiri atar dimensi vertikal (waktu) dan dimensi horizontal (objek-objek yang terkait). *Sequence diagram* biasa digunakan untuk menggambarkan skenario atau rangkaian langkah-langkah yang dilakukan sebagai respons dari sebuah *event* untuk menghasilkan *output* tertentu. Diawali dari apa yang men-*trigger* aktivitas tersebut, proses dan perubahan apa saja yang terjadi secara internal dan *output* apa yang dihasilkan Dharwiyanti (2006).

2.4.2 *Rational Unified Process* (RUP)

Untuk pengembangan aplikasi pembelajaran Bahasa Korea bagi pemula berbasis Android pada tugas akhir ini menggunakan metode pengembangan perangkat lunak *Rational Unified Process* (RUP).

2.4.2.1 Pengertian RUP

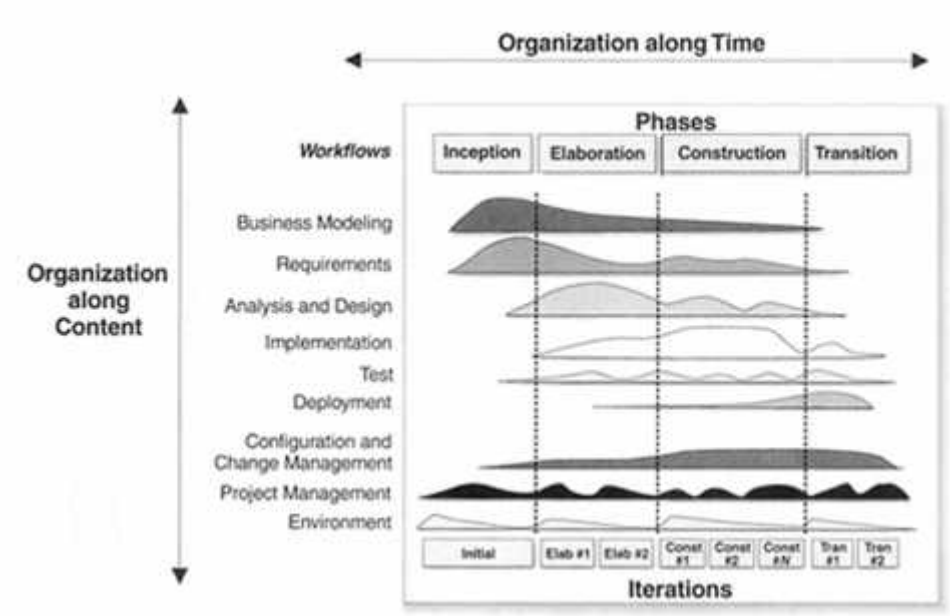
Rational Unified Process adalah sebuah Proses Rekayasa Perangkat Lunak. RUP menyediakan pendekatan disiplin untuk memberikan tugas dan

tanggung jawab dalam organisasi pengembang perangkat lunak. Tujuannya untuk memastikan perangkat lunak yang berkualitas tinggi dan sesuai kebutuhan penggunanya dalam anggaran dan jadwal yang dapat diprediksi (Kruchten, 2000).

RUP mengarahkan kita terhadap pengembanga perangkat lunak secara praktis dan efektif. Terdapat 6 *best practice* atau disebut juga *basic principle* dalam metode RUP, antara lain (Kruchten, 2000):

1. *Develop software iteratively*, bertujuan untuk mengurangi resiko pada awal proyek.
2. *Manage requirements*, bertujuan untuk mengatur kebutuhan yang diperlukan selama proyek.
3. *Use component-based architectures* untuk membangun komponen arsitektur sebuah proyek.
4. *Visually model software*, bertujuan untuk merancang sebuah model visual perangkat lunak, untuk mendapatkan struktur dan perilaku dari aritektur perangkat lunak.
5. *Continuously verify software quality*,
6. *Control changes to software.* kemampuan untuk mengatur serta mengubah perangkat lunak saat dibutuhkan.

RUP menggunakan konsep *object oriented*, dengan aktifitas yang berfokus pada pengembangan model dengan menggunakan *Unified Model Language* (UML). Melalui gambar 2.2 dibawah dapat dilihat bahwa RUP memiliki 2 dimensi, yaitu:



Gambar 2.2 Struktur proses 2 dimensi RUP

Dimensi pertama digambarkan secara horizontal. Dimensi ini mewakili aspek-aspek dinamis dari pengembangan perangkat lunak. Aspek ini dijabarkan dalam tahapan pengembangan atau fase. Setiap fase akan memiliki suatu *major milestone* yang menandakan akhir dari awal dari phase selanjutnya. Setiap phase dapat berdiri dari satu atau beberapa iterasi. Dimensi ini terdiri atas *Inception*, *Elaboration*, *Construction*, dan *Transition*.

Dimensi kedua digambarkan secara vertikal. Dimensi ini mewakili aspek-aspek statis dari proses pengembangan perangkat lunak yang dikelompokkan ke dalam beberapa disiplin. Proses pengembangan perangkat lunak yang dijelaskan kedalam beberapa disiplin terdiri dari empat elemen penting, yakni *who is doing*, *what*, *how* dan *when*. Dimensi ini terdiri atas *Business Modeling, Requirement, Analysis and Design, Implementation, Test, Deployment, Configuration* dan *Change Manegement, Project Management, Environtment.*

2.4.2.2 Fase RUP

Fase-fase pada RUP berdasarkan waktu pengerjaan proyek dapat dibagi menjadi 4 fase, yaitu *Inception, Elaboration, Construction* dan *Transition* (Rational Team, 2001).

1. Fase *Inception*

   Fase *inception* merupakan fase untuk mengidentifikasi masalah, untuk itu diperlukan juga idetifikasi entitas dari luar yang berhubungan dengan sistem. Pada fase ini melibatkan semua identifikasi *use case* dan gambaranya. Selain itu juga termasuk kriteria keberhasilan proyek, perkiraan resiko, perkiraan terhadap *resource* yang dibutuhkan dan merencanakan penjadwalan *milestone*. Hasil yang diperoleh pada fase ini adalah :

   a. Dokumen visi (visi dari kebutuhan projek, kata kunci, batasan utama).
   b. Inisialisasi model *use-case* (10%-20% selesai).
   c. Daftar kata.
   d. *Business case*, termasuk didalamnya konteks bisnis, kriteria sukses, pengenalan pasar dan proyeksi keuangan.
   e. Rencana proyek dan menunjukan fase serta iterasi.
   f. Model bisnis jika diperlukan

   Kriteria evaluasi untuk fase *Inception* adalah :

   a. Menyesuaikan *stakeholder* dengan *scope definition* dan perkiraan biaya atau perkiraan jadwal.
   b. Pemahaman terhadap *use-case* utama
   c. Kredibilitas dari perkiraan biaya, jadwal, prioritas, resiko dan proses pengembangan.
   d. Pemahaman terhadap *prototype*

2. Fase *Elaboration*

   Tujuan dari fase *elaboration* (pengembangan) adalah menganalisa area permasalahan, mengembangkan rencana proyek, dan menghilangkan unsur-

unsur yang memiliki resiko besar terhadap proyek. Adapun hasil dari fase *elaboration* adalah:

a. *Use case* model, seluruh use case dan aktor telah teridentifikasi.
b. *Requirement* tambahan yang mungkin tidak bersifat fungsional bagi proyek.
c. *Software Architecture Description* (Deskripsi Arsitektur Perangkat Lunak).
d. Prototipe dari arsitektur yang dapat dieksekusi.
e. Revisi daftar tingkat resiko dan revisi *business-case*.
f. Rencana pengembangan keseluruhan proyek.
g. Persiapan dokumen panduan bagi pengguna (*user manual*).

Kriteria utama dalam fase elaboration melibatkan pertanyaan berikut :

a. Apakah produk sudah stabil ?
b. Apakah rancangan arsitekturalnya sudah stabil ?
c. Apakah saat demo prototipe, unsur yang memilki resiko telah bisa di atur ?
d. Apakah rencana kontruksi telah detail dan akurat ?
e. Apakah *stakeholder* bersedia dan menyepakati visi dari pengembangan proyek tersebut?
f. Apakah pembelanjaan *actual-resource* terhadap rencana pembelanjaan dapat diterima?

3. Fase *Contruction*

Selama fase kontruksi, semua komponen dan fitur yang dikembangkan terintergrasi ke dalam produk dan secara menyeluruh semua fitur telah diuji. Di lain sisi, proses konstruksi adalah sebuah proses *manufacturing*, dimana terdapat penekanan dalam mengelola *resource* dan mengatur operasi untuk mengoptimalkan jadwal dan kualitas. Pada tahap ini pola pikir (*mindset*) mengalami perubahan dari pengembangan *intellectual property* pada fase

Inception dan *Elaboration*, menjadi pengembangan *deplyoable product*. Kriteria evaluasi terhadap fase *Construction* ini adalah :

a. Apakah peluncuran produk cukup baik dan dapat diterima di komunitas pengguna?
b. Apakah semua *stakeholder* siap untuk beralih ke komunitas pengguna?
c. Apakah pembelanjaan *actual-resource* terhadap rencana pembelanjaan masih tetap diterima?

4. Fase *Transition*

Tujuan dari fase ini adalah untuk transisi dari produk perangkat lunak ke pengguna akhir. Apabila produl telah di luncurkan kepada pengguna, maka isu-isu akan muncul dari pengguna. Nantinya isu ini akan digunakan untuk tahap perbaikan terhadap produk. Kriteria evaluasi untuk fase *Transition* adalah :

a. Apakah pengguna merasa puas?
b. Apakah pembelanjaan *actual-resource* terhadap rencana pembelanjaan masih tetap diterima?